Wireless User Interface Components for Personal Area Networks

Kenneth P. Fishkin, Intel Research Seattle
Kurt Partidge and Saurav Chatterjee, University of Washington

IRS-TR-02-019

October, 2002

Wireless User Interface Components for Personal Area Networks

Wireless wearable systems allow new user interface components. We highlight two—TiltType, a wrist-mounted device for text I/O, and phicons, objects whose physical appearances are metaphors for their electronic capabilities. Artifacts like these can create smaller, simpler, easier wearable systems.

Kenneth P. Fishkin
Intel Research Seattle

Kurt Partridge and Saurav Chatterjee
University of Washington

Although the compelling vision of a "wearable computer" is over 50 years old,[1] it is still only sporadically realized. Some obstacles are transient limits of current technology (weight, cost, and so on) and can be overcome eventually, but other problems are more fundamental and structural. In particular, requiring that wearable computer components be connected via wires has imposed severe constraints on wearable design and acceptance. The recent emergence of cheap, low-power hardware and viable wireless transceivers lets us remove these constraints. Wearable systems composed of wireless modules, known as Wireless Personal Area Networks,[2] are arriving.

Given all the aspects of a wearable system, this seems a rather small change. However, it can have a significant effect on the range of PAN designs. In describing our WPAN, the Spartan BodyNet (SBN), we demonstrate aspects of this new design landscape that let us introduce two new wearable components. The first is TiltType, a wrist-mounted user interface device,[3] and the second are *phicons*, transferable physical icons that convey semantic information.[4] These components have been employed in other user interfaces, but not in a PAN, wired or not.

Why wireless?

When a wearable system is wired, the wires limit the number of components that can be plugged together, the distances between the components, and the number of layers of clothing that can be crossed. Without wires, these constraints disappear. In 1993, Olin Shivers wrote a visionary paper proposing a wireless wearable system that he called "BodyNet,"[5] whose benefits were:

- *Size and location*. When the number of components increases, the responsibility of each decreases. Instead of a one-size-fits-all set of components, each component can tailor itself to one particular task and optimize its shape, size, and location accordingly. For example, medical monitoring components such as heart monitors can be located close to the appropriate area of the body without "dragging" the rest of the system with them, and the shape need only accommodate the monitoring function.
- *Customization*. Because components are specialized and removable, they can also be tailored—users can hot swap user interface components to fit their particular needs or preferences. For example, a user's WPAN might have several different feedback modules (tactile, visual, or auditory) for different contexts (as Rebecca Hansson and her colleagues suggest[6]). The user can tailor each of those modules to the user's preferences.
- *Consolidation*. Components can be easily shared among applications. For example, there is no reason for both a cell phone and pager to contain a list of phone numbers—they should consolidate

1536-1268/02/$17.00 © 2002 IEEE

their information. I/O devices such as headphones, displays, speakers, and microphones could serve any application that needs them.

- *Kinesthetics*. Consumers generally prefer wireless devices because wires can tangle, restrict movement, be tripped over, and get caught on other objects. Devices such as WPAN-participating wristwatches would most likely not be accepted commercially if wired to other wearables.

We propose three additional benefits: unobtrusiveness, multiplicity, and transferability. First, making the system smaller, wireless, and less obvious to other people lessens the "cyborg look" and makes the system less obtrusive. So, the perceived distancing effect of having a wearable system would not come into play as often or as deeply. Second, because we can have many more components, we can readily support multiples of any particular component (for example, multiple user interface components). Because the components don't use physical connectors, they can simultaneously interact. Third, when the components become multiple and wireless, users can capriciously pick them up, set them down, or hand them off, making them more transferable.

When the components become multiple and wireless, users can capriciously pick them up, set them down, or hand them off, making them more transferable.

These benefits can be synergistic and their combinations provide new artifacts and use scenarios. For example, later we combine transferability, customization, and multiplicity benefits into phicons, a new type of UI artifact for WPANs. Similarly, by combining the location, kinesthetics, and unobtrusive benefits, we obtain TiltType, a second new type of user interface artifact for WPANs.

However, compared to their wired counterparts, WPANs do have some disadvantages, the first of which is less available power. Small devices are constrained by their battery size, and practical wireless transfer of power between components is infeasible. Many researchers are investigating ways to reduce power consumption and improve power generation. There is also an association problem.[7] If two PAN components are near each other, are they part of the same PAN? In a wired system, the answer is simple—if they are connected to each other, they are. In a wireless system, the answer is no longer clear—with which PAN is a component associated? This is an active research area, with several solutions under investigation.[8,9]

Another disadvantage is that data rates are reduced. Maximum data rates for wires far outstrip wireless connections. Many applications, however, have modest data rate requirements.

Finally, we need a wireless protocol standard. Wireless protocols have many more constraints than their wired counterparts with respect to range, frequency spectrum, and so on. The Bluetooth wireless communication protocol has recently emerged to address this. It is ideal for WPANs because its nominal range is only 10 meters. It consumes less power and has much less range than other standards that target wireless LANs such as 802.11. Bluetooth is expected to be available in over 70 percent of mobile handsets by 2006.[10] Bluetooth's lower layers have recently been adopted as the IEEE 802.15.1 standard.[2] For our early prototype, we did not use Bluetooth, as it was not quite ready for quick prototyping, but conversion would not be difficult.

So, although there are disadvantages, they are tractable and diminishing, and the benefits outweigh the negatives.

Spartan BodyNet

We wanted to design a small, simple, wireless system that was easily reconfigurable and would let us explore WPAN user interface possibilities. We have used SBN to build a simple prototype application involving stock trading. SBN's four components are a server, phicon, notification ring, and TiltType (see Figure 1).

The server is a 3650 iPAQ, with 32 Mbytes of RAM, 16 Mbytes of ROM, and an optional 802.11b wireless LAN card (used for LAN connections, not the WPAN). The iPAQ's screen, buttons, and stylus are not used in the SBN. The iPAQ could be replaced with a smaller device that contains only the processor, storage, and wireless connectivity, such as the Personal Server.[11]

A phicon is a manipulated object whose physical appearance serves as a metaphor for the object's electronic capabilities. In our sample application, a phicon shaped like a lock gives an additional level of security to the application.

The notification ring's purpose is to attract the user's attention. It has two LEDs, red and yellow, which flash in distinctive patterns to indicate a new event's meaning and priority. For example, the ring could be used to notify users that they are receiving a cell-phone call in a way that is both subtle and public. A touch on the ring lets the user turn notifications off. In our sample application, the notification ring notifies users that the system would like them to enter a stock symbol.

TiltType is a wrist-mounted device for displaying and entering messages. Its current form is the size of a large watch.

University of California at Berkeley's dot motes[12] (see Figure 2) allowed these components to communicate to each other. The motes were originally designed as a minimal hardware device to prototype wireless ad-hoc networks, although we did not use their multihop capabilities for this communication. Each mote contains an Atmel AT90S8535 microcontroller with 512 bytes of RAM, 512 bytes of EEPROM and 8 Kbytes of flash ROM storage. For I/O, each mote has two LEDs, a light sensor, a temperature sensor, and an RF Monolithics TR1000 916.5 MHz short-range radio. A 3V CR2032 lithium coin cell powers the

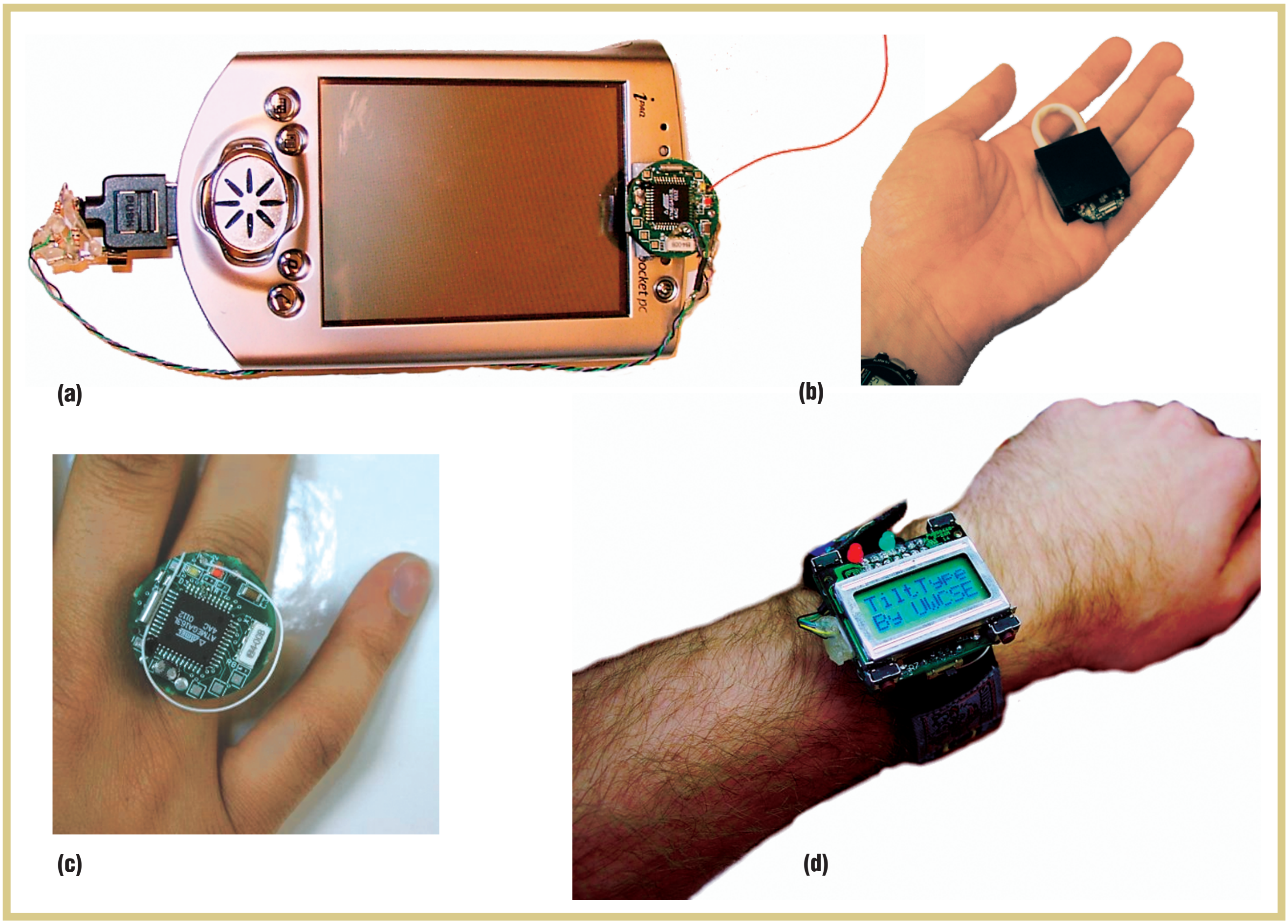


Figure 1. Spartan BodyNet components: (a) the server, (b) a phicon, (c) the notification ring, and (d) TiltType.

mote, which consumes 15 mW of current nominally and about 40 mW while the radio operates. A dot mote is circular, measuring 2.6 cm in diameter and is 8 mm tall including the battery. The server and TiltType connect to motes over serial links, while the functionality of the indicator ring and phicons are implemented directly in a mote's microcontroller.

Dot motes greatly simplified our implementation by providing a small general-purpose radio platform that we were able to implement quickly. Unfortunately, the radio runs at 19.2 kbps. A newer mote can run up to 115 kbps but uses a much larger AA battery pack. Although some wearable applications require faster speeds, 19.2 kbps is adequate for our current research.

The communication protocol between motes uses the mote's standard packet format, which provides addressing and CRC checking. Discovery, the general mechanism by which nodes in a communication network announce themselves, is presently only partially implemented in a non-general way.

TiltType

Figure 3 shows TiltType, a wrist-mounted device that serves as SBN's primary user interface. [3] TiltType gives users the convenience of a wrist-mounted I/O device without the space and power constraints that wrist-mounted computers must endure. TiltType can display short messages while on the user's wrist. To enter text, users

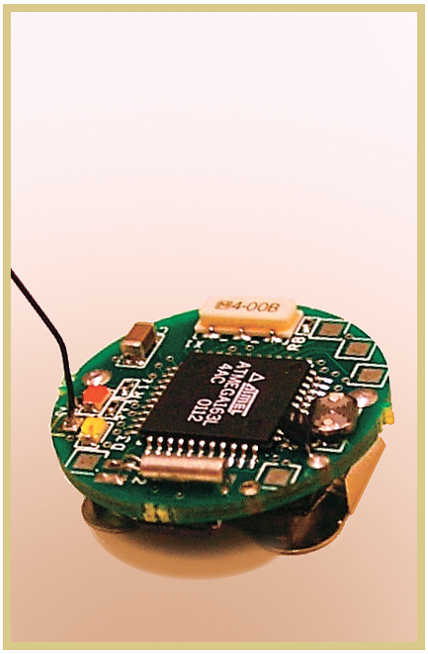


Figure 2. A dot mote.

must remove the device from their wrist because TiltTyping on the wrist is too fatiguing. In our prototype, Velcro keeps TiltType on the wrist, but a snap-on mechanism would be better for a real system. The user then uses a combination of tilts and button presses to enter specific characters (see the TiltType sidebar).

Because of our desire for a cheap, small, easily prototyped display, TiltType's display is only eight characters wide and two lines deep. Although limited, we found it sufficient for delivering short messages and asking the user simple queries. For example, in our sample application, users used TiltType to enter a stock symbol. We augmented the normal text-entry mechanism with a Choice command, which lets users use tilting to select from a range of choices.

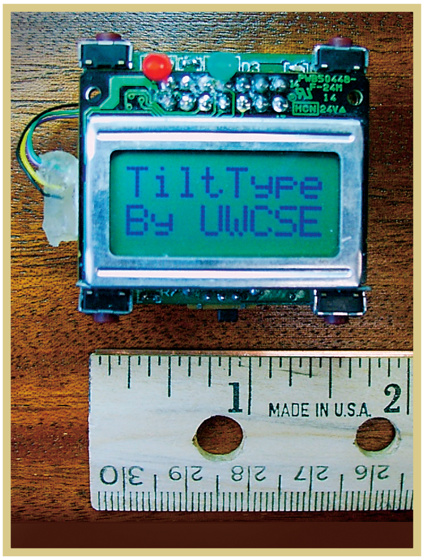


Figure 3. TiltType.

Phicons

The user interface community has recently investigated tangible user interfaces. Their physical appearance suggests natural manipulations with correspondingly intuitive actions, harnessing a metaphor's power to create a more compelling user experience.[4] A real-world example is a car seat adjustment control whose shape resembles a car seat. When the seat control is pressed forward, the seat itself moves forward.

Benefits

Phicons are easy to learn and remember. Also, because they are tied to a particular purpose, they can be faster and less error-prone than general-purpose user interfaces for frequently performed operations.

In addition, their physical appearance can be easily customized. People using phicons might be more likely to customize how they interact with their information; few users customize their user interfaces, yet everybody customizes what they wear and carry.

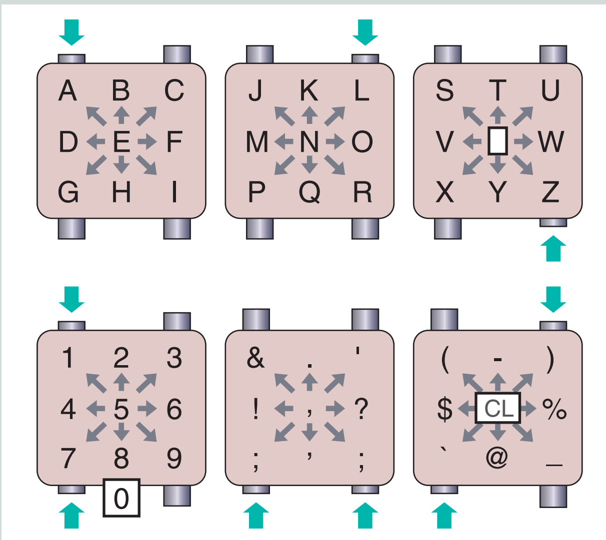
WPAN benefits are particularly synergistic with phicons. Because there are no physical connectors, there is no physical limit to the number of phicons that can interact with a system. WPANs allow the transfer of components, and physical objects such as phicons are also easily transferred from person to person (although they must be reassociated with the new user's WPAN). Although association might present problems, the converse, *dissociation*, presents opportunities because the system can notice when a device leaves a WPAN. For example, in a WPAN, an authorization phicon (discussed later) might be hidden in a person's clothing. When the phicon leaves the WPAN, the WPAN can be notified and can disable access to personal data. That mechanism would let users use other user interfaces found in the environment, such as large public displays,[13] without exposing their data.

Phicons as an aid in authorization

An authorization phicon conceptually resembles a wireless smart card; it is built from tamper-resistant hardware and provides a form of authentication. When the WPAN detects the presence of an authorization phicon, its applications can provide a higher level of functionality. The

TiltType

TiltType users enter characters through a combination of tilting actions and button presses. Nine tilt positions are possible, corresponding to the eight compass directions and keeping the device level. Pressing different buttons while tilting brings up different character maps. With the nine tilting positions, three character maps cover the English alphabet. A fourth button is used for backspace. Simultaneous combinations of the four buttons select additional character maps for numbers, symbols, and special functions. For example, to enter the "i" in "ieee," the user presses the upper left button and tilts TiltType down and to the right.



authorization phicon requires little hardware, so it can be small. Because of this it can be located in housings and places that have the appropriate affordances, such as keys, key rings, earrings, eyeglass frames, cards for wallets, belt buckles, and so on—items that are already personal, easily and frequently worn, carry the appropriate cognitive affordance, and can be augmented comfortably.

For example, in our prototype application, we built an authorization phicon that looks like a lock (see Figure 1). When this phicon is present (the WPAN discovers it), the WPAN allows an additional level of functionality. We implement this using WPAN discovery—when the WPAN discovers the phicon's presence, it conveys its semantics. A real system would require cryptographic security. The phicon can also contain set data, which can be automatically and passively transmitted to the WPAN either upon discovery or in response to some explicit user action.

Authorization levels need not be binary—by having more than one authorization component, or an authorization component with levels, it can communicate a range of authorizations to entities. To expand on the wireless smart-card example, a cell phone might allow calls to 911 only if there is no authorization, or might not allow roaming calls if there is only limited authorization. However, it does allow a call if there is full authorization.

Jalal Al-Muhtadi and his colleagues have recently discussed the concept of storing cryptographic keys in a wearable device,[14] although in the context of a ubiquitous computing environment rather than a WPAN. All the keys are stored in a single place (in the user's watch) rather than being distributed and externally invisible.

Current power technology makes extremely long-lived WPAN phicons impractical, but projected increases in battery power and decreases in radio energy requirements will address this. For example, the 802.15.4 IEEE task force is investigating ways to reduce the power requirements of radio protocols by an order of magnitude over Bluetooth.

Conceptually, Alice giving Bob an authorization phicon is no different from Alice giving Bob some data virtually. However, the use scenario is very different because the capability is now made into a physical object. This is easier for Bob to remember and use, more difficult for others to copy, and is language independent. Of course, this choice inherits the disadvantages of a physical object—the phicon must be stored, is harder to transfer over long physical distances, and so forth. Users can use physical techniques such as phicons alongside virtual techniques such as passwords—the system designer can choose when to employ which. This is another example of how WPANs expand the range of design choices.

Orchestrating the user interface

One advantage of WPANs—the fact that components can dynamically come and go at will—complicates user interface programming. At any time, there is some set of user interface components known to the system. Their capabilities might disjoint, overlap, or even be identical (for example, a user might have multiple "button" widgets scattered all over their body). When the system wishes to perform some UI task, how does it decide which components to notify of that task and what to tell them to do?

> SBN user interface modules, when discovered, announce their unique identifiers, and the host application determines when and how to best interact with that module.

One option is for the system to assume that all user interface components can accommodate all requests, each user interface component being responsible for handling the request as best it can. For example, the system request might be "notify user of a high-priority event." Modules might vibrate, flash, beep, and so on, to accomplish this. While this approach is the simplest for the server, it is not appropriate for WPANs. Different user interface modules have hugely different functionalities; it wastes bandwidth and energy to inform all modules of all tasks. In addition, if the user has many user interface modules on their person, the effect of all of them going off simultaneously could be quite undesirable.

A second option is to have user interface modules send the application a checklist indicating which features they can support (for example, "I can display color, 8 bit, 60 by 80 resolution, and can play audio files in .wav format of <= 64KB"). This eliminates wasted notifications but requires that the set of possible capabilities and the descriptors for each be well understood and regularized. We felt this was not yet appropriate for such a flexible and emerging area.

The third option, which puts the most work on the application but supports the most flexibility, is for the user interface modules to announce their exact ID to the application ("I am a model <X> widget"), and have the application understand the set of module IDs. It could then determine how and when to interact with the module. We use this approach in SBN. SBN user interface modules, when discovered, announce their unique identifiers ("I am a TiltType module"), and the host application determines when and how to best interact with that module.

Sample applications

To show how a simple WPAN like SBN can be used in practice, we demonstrate how our system performs a simple test application. It demonstrates the use of authorization phicons, multiple user interface components, and TiltType. This application involves trading stocks; a user enters a stock symbol, and TiltType displays that stock's current value. If the security phicon

is present, then the user can specify an action to undertake with the stock (buy, sell, stop-loss, and so on). If it is not present, then the stock price alone is displayed.

First, the SBN network discovers whatever phicons are present in the WPAN. In this example, it is informed of the security phicon's existence (the lock in Figure 1). By pressing a certain button sequence using TiltType, the user invokes the stock application.

The application starts by sending a directive to the TiltType module to display a string prompting the entry of a stock symbol (see Figure 4), and to the notification ring (shown in Figure 1) to display a notification that will prompt users to interact with their TiltType module.

Once a user has entered a stock symbol using TiltType, the string is sent from TiltType back to the application as a name–value pair (tag name, entered value). For example, in this case the name–value pair is "stock symbol, INTC."

The application then queries the outside world for the stock's present value and directs TiltType to display it. If the security phicon is not present, the application loops back and prompts the user for more stocks of interest. If the security phicon is present, the application directs TiltType to display a set of choices, one for each action that can be undertaken on the stock (for example, buy, sell, buy-limit, and sell-limit). Using TiltType, the user then selects from the menu of choices, and the selected option is transmitted back to the application.

Other scenarios using phicons

A phicon can also be used in a WPAN as a data reference. For example, when a "Statue of Liberty" charm on a user's charm bracelet is triggered, the WPAN can retrieve the latest New York news from a Web site. WPANs with more multimedia capabilities could support more advanced operations. For example, a smart business card could include an image of a phone that automatically dials the card's phone number when pressed. Advertisers could use phicons to promote their products; when invoked, the phicon could retrieve location information through the WPAN to give directions to the nearest location. The advertiser might later collect the phicons in exchange for a product discount.

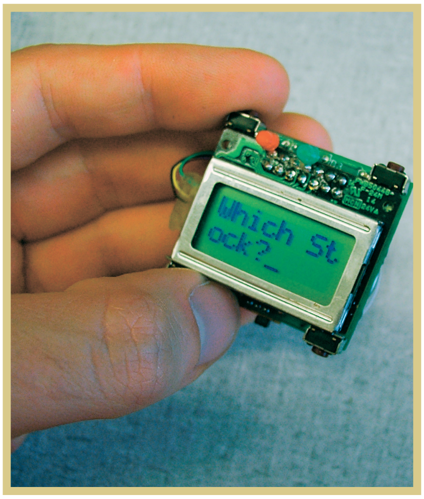


Figure 4. A prompt for entering a stock symbol using TiltType.

In addition, we could use WPAN phicons as a remote-control device. Remote-control devices can operate outside a WPAN, but if in one, users can modify their operation according to their preferences. For example, a miniature light switch might adjust the ambient lighting. Users could control ambient music with the help of a model of their listening preferences, stored on the WPAN server.

Because phicons can contain data, they can let users temporarily lend sensitive information. Suppose Alice wishes to let Bob buy some groceries using her automated teller machine card. If Alice can encode her PIN into a phicon, she can lend that physical artifact to Bob, who can then use it at a phicon-aware ATM. Once Bob returns the phicon to Alice, she does not need to change her PIN as she would have had to otherwise—Bob was able to use the PIN without learning its value.

Other TiltType scenarios

TiltType can assist with entry of passwords or PINs (such as at ATMs). Currently, the user must trust the ATM hardware to not retain it for use by others. Thieves have taken advantage of the general public's trust by setting up fake ATMs and modifying legitimate ones to record passwords.[15] An alternative is for the user to bring their own interface, which they trust to not retain the password. TiltType is particularly suited to this; users can carry it anywhere without special effort because of its small size and because it can easily be placed anywhere on the body. Using a cryptographic challenge-response protocol, the password is not directly communicated to the ATM, but the ATM is satisfied that the user has entered it. On its end, TiltType discards the PIN once the challenge-response is complete.

WPANs containing a TiltType module can be used in conjunction with large public displays. These displays are typically located in areas with significant foot traffic and show items of interest such as upcoming talks. Suppose we wish to augment a display to let users easily download information into their PDA calendars. The display could detect when a user chooses or clicks on a displayed talk, but this is problematic. The displayed area might not be physically reachable, several people might want to click on the same area at the same time, and different user communities might prefer different UIs. Instead, the display could broadcast its event information to all nearby WPANs. To act on that information, the WPAN only needs a minimal UI because the community display already handles most of the information display. The WPAN need only let the user scroll through the list of talks and select the talk of interest. A small UI module such as TiltType can be useful in this scenario.

Hidden input scenario

Another advantage of WPANs is that modules can be scattered all over the body, heedless of intervening clothing layers. This could be advantageous when at an ATM. At present, the user must enter their PIN using a large keypad and be careful of nearby prying eyes. In a WPAN, a user could have a small numeric input module located in their pocket or handbag. The user could enter their PIN by feel, obtaining a greater level of privacy and security.

Our Spartan BodyNet demonstrated two new examples of the type of novel UI artifacts its advantages (and their combinations) allow: phicons and TiltType. Areas for future work include

- *Practicality.* While our system is working, it is only at a prototype level in terms of industrial design. We could dramatically improve the form factors, power requirements (such as incorporating the 802.15.4 effort), display characteristics, and so on.
- *UI intermediary.* It would be interesting to explore more scenarios in which WPAN UI components are used as intermediaries between two "faceless" components, such as a community display, a Personal Server,[11] and so on.
- *Tangible transfer.* We have discussed a few examples of how to integrate the transfer of part of a WPAN from one user to another into the model. We can generalize this in several interesting ways. What if more than one user is involved—for example, when a user needs *k* phicons from *n* other users to be granted some capability? If the granted capabilities are temporally bounded, how are these bounds specified, displayed, and modified? What if these transfers are *n*-way, instead of one-way; for example, if Alice's transfer to Bob changes semantics depending on whether Bob has transferred something to Alice? How do we control transfers of a negative capability, such as "Warning: this bearer is not to be trusted in an e-commerce transaction"?

The Spartan BodyNet is a first step in this direction. We hope it points the way to a future of ubiquitous wearable systems.

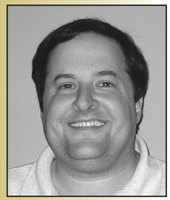
ACKNOWLEDGMENTS

We thank Jason Hill for supporting University of California at Berkeley's dot motes and Gaetano Borriello for his continuing support and counsel.

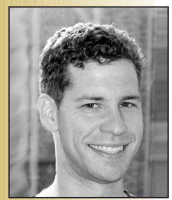
REFERENCES

1. V. Bush, "As We May Think," *The Atlantic Monthly*, July 1945, vol. 176, pp. 101–108.
2. T.M. Siep et al., "Paving the Way for Personal Area Network Standards: An Overview of the IEEE 802.15 Working Group for Wireless Personal Area Networks," *IEEE Personal Comm.*, Feb. 2000, pp. 6–13.
3. K. Partridge et al., "TiltType: Accelerometer-Supported Text-Entry for Very Small Devices," *Proc. Int'l Conf. User Interface Software and Technology* (UIST 02), ACM Press, New York, 2002, pp. 201–204.
4. H. Ishii and B. Ulmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms," *Proc. Conf. Human Factors in Computing Systems* (CHI 97), ACM Press, New York, 1997, pp. 234–241.
5. O. Shivers, "BodyTalk and the BodyNet: A Personal Information Infrastructure," *Personal Information Architecture Note 1*, MIT Laboratory for Computer Science.
6. R. Hannson, P. Ljungstrand, and J. Redstrom, "Subtle and Public Notification Cues for Mobile Devices," *Proc. Int'l Conf. Ubiquitous Computing* (Ubicomp 01), Springer-Verlag, New York, 2001, pp. 240–246.
7. T. Kindberg and A. Fox, "System Software for Ubiquitous Computing," *IEEE Pervasive Computing*, vol. 1, no. 1, Jan./Mar. 2002, pp. 70–81.
8. L.E. Holmquist et al., "Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts," *Proc. Int'l Conf. Ubiquitous Computing* (Ubicomp 01), Springer-Verlag, New York, 2001, pp. 116–122.
9. T. Selker, A. Lockerd, and J. Martinez, "Eye-R, a Glasses-Mounted Eye Motion Detection Interface," *Proc. Conf. Human Factors Computing Systems* (CHI 01), ACM Press, New York, 2001, pp. 179–180.
10. ARC Group, "Bluetooth Personal Area Networks"; www.arcgroup.com/reports/bluetooth/bpan-brochure.pdf
11. R. Want et al., "The Personal Server: Changing the Way We Think about Ubiquitous Computing," *Proc. Int'l Conf. Ubiquitous Computing* (Ubicomp 02), Springer-Verlag, New York, 2002, pp. 194–209.
12. J. Hill et al., "System Architecture Directions for Networked Sensors," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS-X 00), ACM Press, New York, 2000, pp. 93–104.
13. D. Russell and R. Gossweiler, "On the Design of Personal & Communal Large Information Scale Appliances," *Proc. Int'l Conf. Ubiquitous Computing* (Ubicomp 01), Springer-Verlag, New York, 2001, pp. 354–360.
14. J. Al-Muhtadi, D. Mickunas, and R. Campbell, "Wearable Security Services," *Proc. Int'l Conf. Distributed Computing Systems* (ICDCS 21), IEEE CS Press, Los Alamitos, Calif., 2001, pp. 266–271
15. *Boston Globe*, "Fake ATM Plays Gotcha with Users," 12 May 1993.

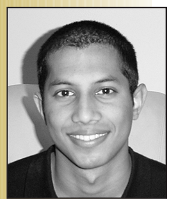
the AUTHORS



Kenneth P. Fishkin is a researcher at Intel Research Seattle. He has more than 15 years experience in industrial research and development. He cofounded the eXtreme UI research effort while at PARC, which helped invent tangible user interfaces. His research interests are ubiquitous computing user interfaces and systems. He holds bachelor's degrees in computer science and mathematics from the University of Wisconsin, Madison, and a master's in computer science from the University of California at Berkeley. He is a member of the ACM. Contact him at Intel Research Seattle, 1100 NE 45th St., Sixth floor, Seattle WA 98105; kfishkin@intel-research.net.



Kurt Partridge is a PhD candidate in the University of Washington's Computer Science and Engineering Department. His research interests include technology to support novel user interaction mechanisms for wearable and ubiquitous computing devices. He received a BS in computer science from the University of California at Berkeley and an MS in computer science from the University of Washington. He is a member of the IEEE and the ACM. Contact him at University of Washington's Department of Computer Science & Engineering, Box 352350, Seattle, WA 98195-2350; kepart@cs.washington.edu.



Saurav Chatterjee is an undergraduate student in the computer science and engineering department and the Honors College at the University of Washington. Previously, he worked with Kurt Partridge to develop TiltType. His research interests include ubiquitous computing and artificial intelligence. He is a member of the IEEE. Contact him at sauravc@u.washington.edu.